(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 557 788 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
27.07.2005 Bulletin 2005/30

(51) Int Cl.[7]: **G06N 3/12**

(21) Application number: **04010194.1**

(22) Date of filing: **29.04.2004**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PL PT RO SE SI SK TR
Designated Extension States:
AL HR LT LV MK

(30) Priority: **26.01.2004 EP 04001615**

(71) Applicant: **Honda Research Institute Europe GmbH**
63073 Offenbach/Main (DE)

(72) Inventors:
- **Yaochu, Jin**
  63073 Offenbach/Main (DE)
- **Sendhoff, Bernhard**
  63073 Offenbach/Main (DE)

(74) Representative: **Rupp, Christian, Dipl.Phys. et al**
Mitscherlich & Partner
Patent- und Rechtsanwälte
Sonnenstrasse 33
80331 München (DE)

(54) Reduction of fitness evaluations using clustering technique and neural network ensembles

(57) The invention proposes an evolutionary optimization method. In a first step, an initial population of individuals is set up and an original fitness function is applied. Then the offspring individuals having a high evaluated quality value as parents are selected. In a third step, the parents are reproduced to create a plurality of offspring individuals. The quality of the offspring individuals is evaluated by means of a fitness function, wherein selectively the original or an approximate fitness function is used. Finally, the method goes back to the selection step until a termination condition is met.

The step of evaluating the quality of the offspring individuals consists in grouping all λ offspring individuals in clusters, selecting for each cluster one or a plurality of offspring individuals, resulting in altogether ξ selected offspring individuals, evaluating the ξ selected offspring individuals by means of the original fitness function, and evaluating the remaining λ-ξ offspring individuals by means of the approximate fitness function.

S31. Initialize λ individuals, evaluate all individuals using the original fitness function

S32. For each generation:

S32a. select the best μ individuals

S32b. generate λ offspring individuals by recombination and mutation

S32c. evaluate
- clustering the λ individuals using the k-means algorithm (S32d),
- select the ξ individuals closest to the cluster centers (S32e),
- evaluate the ξ selected individuals of step S32e using the original fitness function (S32f),
- construct the neural network ensemble (S32g), and
- calculate the fitness of the remaining λ-ξ individuals using the neural network ensemble. (S32h)
- for the λ-ξ individuals, calculate the variance of output of the ensemble members. If the variance is larger than a given threshold, replace the fitness value with that of the individual closest to the cluster center

S33. Go to step S32 if the termination condition is not met

S34. Stop

Evolutionary optimization algorithm

Fig. 3

EP 1 557 788 A2

Description

FIELD AND BACKGROUND OF THE INVENTION

[0001] The underlying invention generally relates to the field of evolutionary computation, especially to the reduction of the number of fitness evaluations in evolutionary algorithms (EA).

[0002] Many difficulties can arise in applying evolutionary algorithms to solve complex real-world optimization problems. One of the main concerns is that evolutionary algorithms usually need a large number of fitness evaluations to obtain a good solution. Unfortunately, fitness evaluations are often time-consuming. Taking aerodynamic design optimization as an example, one evaluation of a given design based on the 3-Dimensional Computational Fluid Dynamics (CFD) Simulation can take hours even on a high-performance computer.

BRIEF DESCRIPTION OF THE PRESENT STATE OF THE ART

[0003] The background art references used in the present description are summarized and commented on in a table at the end of the description.

[0004] Whenever in the present specification reference is made to the background art (in brackets), it is to be understood that the corresponding document is incorporated thereby by reference.

[0005] To alleviate the fact that evolutionary computation is often time consuming, computationally efficient models can be set-up to approximate the fitness function. Such models are often known as approximate models, meta-models or surrogates (see [8] for an overview of this topic). It would be ideal if an approximate model can fully replace the original fitness function, however, research has shown that it is in general necessary to combine the approximate model with the original fitness function to ensure the evolutionary algorithm to converge correctly. To this end, re-evaluation of some individuals using the original fitness function, also termed as evolution control in [6], is essential.

[0006] Generation-based or individual-based evolution control can be implemented. In the generation-based approach [14, 2, 6, 7], some generations are evaluated using the approximate model and the rest using the original fitness function. In individual-based evolution control, part of the individuals of each generation are evaluated using the approximation model and the rest using the original fitness function [6, 3, 17, 1]. Generally speaking, the generation-based approach is more suitable when the individuals are evaluated in parallel, where the duration of the optimization process depends to a large degree on the number of generations needed. By contrast, the individual-based approach is better suited when the number of evaluations is limited, for example, when an experiment needs to be done for a fitness evaluation.

[0007] On the other hand, individual-based èvolution control provides more flexibility in choosing which individuals need to be re-evaluated. In [6] it is suggested to choose the best individuals according to the approximate model rather than choosing the individuals randomly. In [3] not only the estimated function value but also the estimation error are taken into account. The basic idea is that individuals having a larger estimation error are more likely to be chosen for re-evaluation. Other uncertainty measures have also been proposed in [1].

[0008] In [9] the population of a genetic algorithm (GA) is grouped into a number of clusters and only one representative individual of each cluster is evaluated using the fitness function. Other individuals in the same cluster are estimated according to their Euclidean distance to the representative individuals. Obviously, this kind of estimation is very rough and the local feature of the fitness landscape is completely ignored.

OBJECT OF THE UNDERLYING INVENTION

[0009] In view of the prior art, it is the object of the present invention to propose a computationally efficient model for fitness evaluations to assist the evolutionary algorithms.

[0010] This object is achieved by means of the features of the independent claims. Advantageous features are defined in the dependent claims.

SUMMARY OF THE INVENTION

[0011] In the proposed invention, the population is grouped into a number of clusters, and only the individual that is closest to the cluster center is evaluated using the original fitness function. In contrast to the distance-based estimation method [9], according to a further aspect of the invention the evaluated individuals (centers of the clusters) can be used to create a neural network ensemble, which is then used for estimating the fitness values of the remaining individuals. Both the structure and the parameters of the neural networks can e.g. be optimized using an evolutionary algorithm (EA) with Lamarckian inheritance.

[0012] To reduce the number of fitness evaluations, the present invention thus proposes a computationally efficient model that can be constructed for fitness evaluations to assist the evolutionary algorithms.

[0013] The invention suggests a method for reducing the required number of fitness evaluations using meta-models. It deals with three aspects in meta-model assisted evolutionary computation.

[0014] First, how to determine which individuals should be evaluated with the original time-consuming fitness function and which ones should be evaluated using the meta-model: The invention uses a clustering method using a k-means algorithm.

[0015] Second, how to improve the quality of the meta-models. A neural network ensemble has been used instead of a single neural network. Meanwhile, both the structure and parameters of the neural networks are on-line optimized using a Lamarkian evolutionary algorithm. Finally, the weights of the ensemble members are optimized using a standard Evolution Strategy (ES).

[0016] Third, how to detect a large prediction error of the meta-models. The invention uses the output variance of the ensemble members.

BRIEF DESCRIPTION OF THE CLAIMS

[0017] According to a first aspect of the present invention an evolutionary optimization method is proposed. In a first step, an initial population of individuals is set up and an original fitness function is applied. Then the offspring individuals having a high evaluated quality value are selected as parents. In a third step, the parents are reproduced to create a plurality of offspring individuals. The quality of the offspring individuals is evaluated by means of a fitness function, wherein selectively the original or an approximate fitness function is used. Finally, the method goes back to the selection step until a termination condition is met. The step of evaluating the quality of the offspring individuals further consists in grouping all λ offspring individuals in clusters, selecting for each cluster one or a plurality of offspring individuals, resulting in altogether ξ selected offspring individuals, evaluating the ξ selected offspring individuals by means of the original fitness function and evaluating the remaining λ-ξ offspring individuals by means of the approximate fitness function.

[0018] The approximate fitness function can e.g. be a neural network ensemble comprising a plurality of neural network.

[0019] The offspring individuals chosen for evaluation with the original fitness function can e.g. be the one closest to the cluster center.

[0020] A k-means method can be used to group all offspring into clusters.

[0021] Moreover, the output of each neural network in the ensemble can be weighted and combined to the final output of the neural network ensemble.

[0022] A Genetic Algorithm (GA) with a local search can also be used to generate the neural network ensemble.

[0023] An evolution strategy (ES) can be used to optimize the weights of the ensemble by minimizing an expected prediction error.

[0024] The variance of the neural network's output in the ensemble can be derived and if the variance is larger than a given threshold, the corresponding fitness value of the individual concerned can be replaced by the value of the individual closest to the cluster center.

[0025] Said variance can be used to control how many individuals in each cluster are evaluated using the original function.

[0026] The number of individuals to be evaluated using the original fitness function can be increased if the variance is high and decreased if the variance is low.

[0027] According to another aspect of the invention a computer software program product implementing a method as set forth above when running on a computing device is proposed.

[0028] According to a further aspect of the invention a use of the presented evolutionary optimization method is proposed for the optimization of hydrodynamic or aerodynamic designs, of turbine rotor and stator blades and outlet guide vanes, and of aerodynamic properties of vehicles.

BRIEF DESCRIPTION OF THE DRAWINGS

[0029] Further advantages of the underlying invention result from the subordinate claims as well as from the following description of two preferred embodiments of the invention which are depicted in the following drawings. Herein,

Fig. 1 shows a known k-means algorithm for population clustering,

Fig. 2 shows an algorithm for constructing neural network ensemble according to the underlying invention,

Fig. 3 shows the evolutionary optimization algorithm proposed by the invention,

Fig. 4 shows a global silhouette width when the number of cluster is set to 10 and the population size is 30 on the 30-dimensional Ackley function,

Fig. 5 shows an ensemble output with an ensemble size of 3 (a) and an ensemble size of 5 (b),

Fig. 6 shows a prediction error versus the standard deviation (a) and a prediction error of the BEM versus that of the GEM (b),

Fig. 7 shows a box plot of the optimization results for the 30-dimensional Ackley function for the proposed algorithm (a) and for Plain ES (b), wherein the scales in (a) and (b) are not the same,

Fig. 8 shows a box plot of the optimization results for the 30-dimensional Rosenbrock function for the proposed algorithm (a) and for Plain ES (b),

Fig. 9 shows a box plot of the optimization results for the 30-dimensional Sphere function for the proposed algorithm (a) and for Plain ES (b), wherein the scales in (a) and (b) are not the same,

Fig. 10 shows the results for the 30-dimensional Ackley function with a single network,

Fig. 11 shows the results for the 30-dimensional Rosenbrock function with a single network,

Fig. 12 shows the results for the 30-dimensional SPH function with a single network.

DETAILED DESCRIPTION OF THE UNDERLYING INVENTION

[0030] When approximate models are involved in evolution, it is necessary to determine which individuals should be re-evaluated using the original fitness function to guarantee a proper convergence of the evolutionary algorithm.

[0031] According to a first embodiment of the invention, the k-means method is applied to group the individuals of a population into a number of clusters. For each cluster, e.g. only the individual that is closest to the cluster center is evaluated using the expensive original fitness function. The fitness of the other individuals is estimated using a neural network ensemble, which is also used to detect possible serious prediction errors. Simulation results from three test functions show that the proposed invention exhibits better performance than the strategy where only the best individuals according to the approximate model are re-evaluated.

Population clustering using the k-means algorithm

[0032] A variety of clustering techniques have been proposed for grouping similar patterns (data items) [4]. All of these clustering techniques can be used in conjunction with the present invention. Generally, they can be divided into:

- hierarchical clustering algorithms, and
- partitional clustering algorithms.

[0033] A hierarchical algorithm yields a tree structure representing a nested grouping of patterns, whereas a partitional clustering algorithm generates a single partition of the patterns.

[0034] Among the partitional clustering methods, the k-means is the simplest and the most commonly used clustering algorithm. It employs the squared error criterion and its computational complexity is O(n), where n is the number of patterns.

[0035] Fig. 1 shows an algorithm for population clustering which is known as such. A typical stopping criterion is that the decrease in the squared error is minimized.

[0036] A major problem of the k-means clustering algorithm is that it may converge to a local minimum if the initial partition is not properly chosen. Besides, the number of clusters needs to be specified beforehand, which is a general problem for partitional clustering algorithms [4].

[0037] To assess the validity of a given cluster, the silhouette method [16] can be used. For a given cluster, X_j, j=l,..., k , the silhouette technique assigns the i-th member (x_{ij}, i=1,...,n_j) of cluster X_j a quality measure (silhouette width):

$$s_{ij} = \frac{b_i - a_i}{\max\{a_i, b_i\}} \qquad \text{(Eq. 1)}$$

where a_i is the average distance between x_{ij} and all other members in X_j and b_i denotes the minimum of a_i, i=1,2,..., n_j, where n_j is the number of patterns in cluster X_j and naturally, $n_1+...+n_k$ equals n if each pattern belongs to one and only one cluster, n is the number of patterns to be clustered.

[0038] It can be seen that s_{ij} has a value between -1 and 1. If s_{ij} equals 1, it means that s_{ij} is in the proper cluster. If s_{ij} is 0, it indicates that x_{ij} may also be grouped in the nearest neighboring cluster and if x_{ij} is -1, it suggests that x_{ij} is very likely in the wrong cluster. Thus, a global silhouette width can be obtained by summing up the silhouette width of all patterns:

$$S = \frac{1}{k} \sum_{j=1}^{k} \sum_{i=1}^{n_j} s_{ij}. \qquad \text{(Eq. 2)}$$

[0039] Consequently, this value can be used to determine the proper number of clusters.

<u>Construction of neural network ensembles using an evolutionary algorithm</u>

[0040] After the population is grouped into a number of clusters, only the individual that is closest to each cluster center is evaluated using the original fitness function. Note that also a plurality of individuals being more or less in the center of a cluster can be selected.

[0041] In [9], the fitness value of all other individuals are estimated based on their Euclidean distance to the cluster center. Obviously, this simplified estimation ignores the local feature of the fitness landscape which can be extracted from the evaluated cluster centers.

[0042] In [6, 7], a standard neural network has been constructed using the data generated during optimization. The neural network model is trained off-line and further updated when new data are available. One problem that may occur is that as the number of samples increases, the learning efficiency may decrease. To improve the learning efficiency, weighted learning [7] and off-line structure optimization of the neural networks have been shown to be promising.

[0043] According to the invention, the approximation quality can be further improved in two aspects:

- First, structure optimization of the neural network is carried out on-line and only the data generated in the most recent two generations are used (S21). This makes it possible to have an approximate model that reflects the local feature of the landscape.
- Second, an ensemble instead of a single neural network is used to improve the generalization property of the neural networks (S22).

[0044] The benefit of using a neural network ensemble originates from the diversity of the behavior of the ensemble members on unseen data. Generally, diverse behavior on unseen data can be obtained via the following approaches:

- Using various initial random weights.
- Varying the network architecture.
- Employing different training algorithms.
- Supplying different training data by manipulating the given training data.
- Generating data from different sources.
- Encouraging diversity [12], decorrelation [15] or negative correlation [10, 11] between the ensemble members.

[0045] In the present invention, a Genetic Algorithm (GA) can be used to generate the neural network ensemble (S22), which can provide two sources of diversity: both the architecture and the final weights of the neural networks are different. Since the goal of the neural networks is to learn the local fitness landscape, the optimization method of the invention only uses the data generated in the two most recent generations instead of using all data.

[0046] Assuming that the λ individuals in the population are grouped into ξ clusters, thus ξ new data will be generated in each generation. Accordingly, the fitness function for evolutionary neural network generation can be expressed as follows:

$$F = \frac{1}{\xi}\left\{\alpha \cdot \sum_{i=1}^{\xi}(y_i - y_i^d(t))^2 + (1-\alpha)\cdot \sum_{i=1}^{\xi}(y_i - y_i^d(t-1))^2\right\}$$

(Eq. 3)

where $0.5 \le \alpha \le 1$ is a coefficient giving more importance to the newest data, $y_i^d(t)$, $i=1,\ldots,\xi$ are the data generated in the current generation and $y_i^d(t-1)$, $i=1,\ldots,\xi$ are those generated in the last generation and y_i is the network output for the i-th data set.

[0047] Given N neural networks, the final output of the ensemble can be obtained by averaging the weighted outputs of the ensemble members:

$$y^{EN} = \sum_{k=1}^{N} w^{(k)} y^{(k)}$$

(Eq. 4)

where $y^{(k)}$ and $w^{(k)}$ are the output and its weight of the k-th neural network in the ensemble. In this case, the expected error of the ensemble is given by:

$$E^{EN} = \sum_{i=1}^{N}\sum_{j=1}^{N} w^{(i)} w^{(j)} C_{ij}$$

(Eq. 5)

where C_{ij} is the error correlation matrix between network i and network j in the ensemble (S23):

$$C_{ij} = E[(y_i - y_i^d)(y_j - y_j^d)]$$

(Eq. 6)

where E(.) denotes the mathematical expectation.

[0048] It has been shown [13] that there exists an optimal set of weights that minimizes the expected prediction error of the ensemble:

$$w^{(k)} = \frac{\sum_{j=1}^{N}(C_{kj})^{-1}}{\sum_{i=1}^{N}\sum_{j=1}^{N}(C_{ij})^{-1}}$$

(Eq. 7)

where $1 \le i,j,k \le N$.

[0049] However, a reliable estimation of the error correlation matrix is not straightforward because the prediction errors of different networks in an ensemble are often strongly correlated.

[0050] A few methods have been proposed to solve this problem [5, 18, 19]. Genetic programming is applied to the search for an optimal ensemble size in [19] whereas the recursive least-square method is adopted to optimize the weights in [18]. In [18], a Genetic Algorithm (GA) is also used to search for an optimal subset of the neural networks in the final population as ensemble members.

[0051] A canonical Evolution Strategy (ES) can be employed to find the optimal weights (S24) to minimize the expected error in Eq. 5.

[0052] The algorithm for constructing the neural network ensemble and the entire evolutionary optimization algorithm are sketched in Fig. 2 and Fig. 3, respectively.

Experimental Setup

[0053] In the simulations, optimization runs are carried out on three well known test functions, the Ackley function, the Rosenbrock function and the Sphere function.

[0054] The dimension of the test functions are set to 30. A standard (5, 30) Evolution Strategy (ES) is used in all simulations.

[0055] To implement the evolutionary optimization with approximate fitness models, a few important parameters have to be determined, such as the number of clusters and the number of neural networks in the ensemble.

[0056] The first issue is the number of clusters. This number is relevant to performance of the clustering algorithm, the quality of the approximate model, and eventually the convergence property of the evolutionary algorithm.

[0057] A few preliminary optimization runs are carried out with only a single neural network being used for fitness approximation on the 30-dimensional Ackley function. It is found that with the clustering algorithm, the evolutionary algorithm is able to converge correctly when about one third of the population is re-evaluated using the original fitness function. When the number of the re-evaluated individuals is much fewer than one third of the population, the performance of the evolutionary algorithm becomes unpredictable, that is, the evolutionary algorithm may converge to a false minimum.

[0058] The clustering performance is then evaluated when the number of clusters is set to be one third of the population. Fig. 4 shows the global silhouette width when the cluster number is 10 and the population size is 30 on the 30-dimensional Ackley function. It can be seen that the clustering performance is acceptable.

[0059] Next, simulations are conducted to investigate the ensemble size. So far, the ensemble size has been determined heuristically in most applications. In [19], the optimal size turns out to be between 5 and 7. Considering the fact that a large ensemble size will increase computational cost, two cases are compared, where the ensemble size is 3 and 5 on 200 samples collected in the first 20 generations of an optimization run on the 30-dimensional Ackley function.

[0060] The ensemble output versus that of a single network is plotted in Fig. 5, where in Fig. 5(a) the ensemble size is 3 and in Fig. 5(b) the ensemble size is 5. Note that the more points locate in the right lower part of the figure the more effective the ensemble. It can be seen from the figure that no significant performance improvement has been achieved when the ensemble size is changed from 3 to 5. Thus, the ensemble size is fixed to 3.

[0061] It seems that the use of an ensemble has not improved the prediction accuracy significantly. Thus, the motivation to employ an ensemble becomes questionable. The following paragraph shows that an ensemble is important not only in that it is able to improve prediction.

[0062] The equally important reason for introducing the ensemble in the optimization according to the invention is to estimate the prediction accuracy based on the different behaviors of the ensemble members, i.e., the variance of the members in the ensemble. To demonstrate this, Fig. 6(a) shows the relationship between the standard deviation of the predictions of the ensemble members and the estimation error of the ensemble. These data are also collected in the first 20 generations of an evolutionary run of the Ackley function. Additional function evaluations are carried out to get the prediction error. Of course, they are neither used in neural network training nor in optimization. It can be seen that a large standard deviation most probably indicates a large prediction error, although a small standard deviation does not guarantee a small prediction error. Encouraged by this close correlation between a large deviation and a large prediction error, a try to predict the model error is made. When the standard deviation is larger than a threshold (1 in this example), the model prediction is replaced by the fitness of the individual closest to the cluster center, which is a very rough but feasible approximation.

[0063] Finally, a standard Evolution Strategy (ES) with a population size of (3,15) is used to optimize the weights of the ensemble members. The predictions of the Generalized Ensemble Method (GEM), where the weights are optimized, and that of a Basic Ensemble Method (BEM) are shown in Fig. 6(b). It can be seen that the prediction accuracy has been improved using the GEM.

Optimization Results

[0064] The evolutionary optimization method of the invention is applied to the optimization of three functions: Ackley function, the Rosenbrock function and the Sphere function. The maximal number of fitness evaluations is set to 2000 in all simulations.

[0065] Fig. 7, 8 and 9 show the box plots of the ten runs on the three test functions. For clarity, only 20 data points are shown in the figures, which are uniformly sampled from the original data. From these figures, it can clearly be seen that on average, the optimization results using the proposed algorithm are much better than those from the plain evolution strategy on all test functions. Meanwhile, they are also much better than the results reported in [6], where no clustering of the population has been implemented. As mentioned, without clustering, the evolutionary algorithm does not converge correctly if only one third of the population is re-evaluated using the original fitness function. Nevertheless, it can noticed that for the Ackley function, the result from one of the 10 runs using the proposed method is much worse than the average performance, even a little worse than the average result when the plain ES is used, refer to Fig. 7(a).

[0066] Fig. 10, 11 and 12 depict the box plots of results using only a single neural network (where no remedy of large prediction errors is included) on the three test functions in order to show the benefit of using the neural network ensemble. Similarly, only 20 data points are presented for the clarity of the figures. Compared with the results shown in Fig. 7, 8 and 9, they are much worse. In the Rosenbrock function, some runs even have diverged, mainly due to the bad performance of the model prediction.

SUMMARY OF THE ADVANTAGES OF THE PRESENT INVENTION

[0067] A new method for reducing fitness evaluations in evolutionary computation is proposed. In each generation, the population is clustered into a number of groups and only the individuals closest to each cluster center is evaluated. Then a neural network ensemble is constructed using the data from the evaluated individuals. To further improve the prediction quality, the weights of the ensemble are optimized using a standard ES.

[0068] The invention further exploit information contained in the ensemble by taking advantage of the standard deviation of the output of the ensemble members. When the ensemble members disagree significantly, the prediction error is very likely to be large and thus the ensemble prediction is replaced by the fitness value of the cluster center of the individual. Simulation results on the test functions suggest that the proposed algorithm is very promising.

[0069] Currently, the number of individuals to be controlled is fixed. As suggested in [7], an adaptation of the control frequency could provide more performance improvement. One possibility is to determine the number of individuals to optimize the performance of the clustering algorithm using the global silhouette width.

[0070] Aspects of the invention are:

- The k-means clustering is used for selecting individuals for re-evaluation in the context of the individual-based evolution control.

- An ensemble, instead of a single model is used to improve the prediction quality. Besides, an ES is used to optimize the weights of the ensemble on-line based on an estimated prediction error.

- One of the main contributions of the invention is that the variance of the ensemble members is exploited to detect large prediction errors. Once such an error is detected, the prediction of the meta-model is discarded and the fitness of the concerned individual is replaced by that of the individual closest to the cluster center.

[0071] The following table summarizes the abbreviations used in the present description:

BEM	Basic Ensemble Method
CFD	Computational Fluid Dynamics
ES	Evolution Strategy
GA	Genetic Algorithm
GEM	Generalized Ensemble Method
SPH	Sphere test function

[0072] The following table summarizes the background art referenced in the present description:

[1]	J. Branke and C. Schmidt. Fast convergence by means of fitness estimation. Soft Computing, 2003.
[2]	M.A. El-Beltagy, P.B. Nair, and A.J. Keane. Metamodeling techniques for evolutionary optimization of computationally expensive problems: promises and limitations. In Proceedings of Genetic and Evolutionary Conference, pages 196-203, Orlando, 1999. Morgan Kaufmann.
[3]	M. Emmerich, A. Giotis, M. Ozdenir, T. Bäck, and K. Giannakoglou. Metamodel assisted evolution strategies. In Parallel Problem Solving from Nature, number 2439 in Lecture Notes in Computer Science, pages 371-380. Springer, 2002.

(continued)

[4]	A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. ACM Computing Surveys, 31(3):264-323, September 1999.
[5]	D. Jimenez. Dynamically weighted ensemble neural networks for classification. In Proceedings of International Joint Conference on Neural Networks, pages 753-756, Anchorage, 1998. IEEE Press.
[6]	Y. Jin, M. Olhofer, and B. Sendhoff. On evolutionary optimization with approximate fitness functions. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 786-792. Morgan Kaufmann, 2000.
[7]	Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. IEEE Transactions on Evolutionary Computation, 6(5):481-494, 2002.
[8]	Y. Jin and B. Sendhoff. Fitness approximation in evolutionary computation - A survey. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 1105-1112, 2002.
[9]	H.-S. Kim and S.-B. Cho. An efficient genetic algorithms with less fitness evaluation by clustering. In Proceedings of IEEE Congress on Evolutionary Computation, pages 887-894. IEEE, 2001.
[10]	Y. Liu and X. Yao. Negatively correlated neural networks can produce best ensemble. Australian Journal of Intelligent Information Processing System, 4 (3-4) :176-185, 1997.
[11]	Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. IEEE Transactions on Evolutionary Computation, 4(4):380-387, 2000.
[12]	D.W. Opitz and J. W. Shavlik. Generating accurate and diverse members of a neural network ensemble. In Advances in Neural Information Processing Systems, volume 8, pages 535-541, Cambridge, MA, 1996. MIT Press.
[13]	M.P. Perrone and L.N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, Artificial Neural Networks for Speech and Vision, pages 126-142. Chapman & Hall, London, 1993.
[14]	A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In A. Eiben, Th. Back, M. Schoenauer, and H.-P. Schwefel, editors, Parallel Problem Solving from Nature, volume V, pages 87-96, 1998.
[15]	B. E. Rosen. Ensemble learning using decorrelated neural networks. Connection Science, 8(3-4):373-384, 1996.
[16]	P.J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20:53-65, 1987.
[17]	H. Ulmer, F. Streicher, and A. Zell. Model-assisted steady-state evolution strategies. In Proceedings of Genetic and Evolutionary Computation Conference, LNCS 2723, pages 610-621, 2003.
[18]	X. Yao and Y. Liu. Making use of population information in evolutionary artificial neural networks. IEEE Transactions on Systems, Man, and Cybernetics-Part B:Cybernetics, 28(3):417-425, 1998.
[19]	B.-T. Zhang and J.G. Joung. Building optimal committee of genetic programs. In Parallel Problem Solving from Nature, volume VI, pages 231-240. Springer, 2000.

Claims

1. An evolutionary optimization method, comprising the following steps:

- setting up an initial population of individuals and applying an original fitness function (S31),
- selecting the offspring individuals having a high evaluated quality value as parents (S32a),
- reproducing the parents to create a plurality of offspring individuals (S32b),
- evaluating the quality of the offspring individuals by means of a fitness function, wherein selectively the original or an approximate fitness function is used (S32c), and
- going back to the selection step (S32a) until a termination condition is met (S33),

wherein the evaluation step (S32c) consists in:

- grouping all λ offspring individuals in clusters (S32d),
- selecting for each cluster one or a plurality of offspring individuals, resulting in altogether ξ selected offspring individuals (S32e),
- evaluating the ξ selected offspring individuals by means of the original fitness function (S32f), and
- evaluating the remaining λ-ξ offspring individuals by means of the approximate fitness function (S32h).

2. A method according to claim 1, wherein the approximate fitness function is a neural network ensemble comprising a plurality of neural network (S32h).

3. A method according to anyone of the preceding claims, wherein the offspring individuals chosen for evaluation with the original fitness function are the one closest to the cluster center (S32e).

4. A method according to anyone of the preceding claims, wherein a k-means method is used to group all offspring into clusters (S32d).

5. A method according to any claims 2 to 4, wherein the output of each neural network in the ensemble is weighted (S24) and combined (S32g) to the final output of the neural network ensemble.

6. A method according to any claims 2 to 5, wherein a Genetic Algorithm (GA) with a local search is used to generate the neural network ensemble (S22).

7. A method according to any claims 2 to 6, wherein an evolution strategy is used to optimize the weights of the ensemble by minimizing an expected prediction error (S24).

8. A method according to any claims 2 to 7, wherein the variance of the neural network's output in the ensemble is derived and if the variance is larger than a given threshold, the corresponding fitness value of the individual concerned is replaced by the value of the individual closest to the cluster center, which has been evaluated using the original fitness function

9. A computer software program product, implementing a method according to any of the preceding claims when running on a computing device.

10. Use of a method according to any claims 1 to 8, for the optimization of hydrodynamic or aerodynamic designs.

11. Use of a method according to any claims 1 to 8, for the optimization of turbine rotor and stator blades and outlet guide vanes.

12. Use of a method according to any claims 1 to 8, for the optimization of aerodynamic properties of vehicles.

S11.	Choose k patterns randomly as the cluster centers
S12.	Assign each pattern to its closest cluster center
S13.	Recompute the cluster center using the current cluster members
S14.	If the convergence criterion is not met, go to step S12; otherwise stop

State-of-the-art k-means algorithm

Fig. 1

S21.	Prepare the training and test data
S22.	Generate N (ensemble size) neural networks using Genetic Algorithm (GA)
S23.	Calculate the error correlation between the ensemble members
S24.	Determine the optimal weight for each network by using Evolution Strategy (ES)

Algorithm for constructing neural network ensemble

Fig. 2

S31. Initialize λ individuals, evaluate all individuals using the original fitness function

S32. For each generation:

S32a. select the best μ individuals

S32b. generate λ offspring individuals by recombination and mutation

S32c. evaluate
- clustering the λ individuals using the k-means algorithm (S32d),
- select the ξ individuals closest to the cluster centers (S32e),
- evaluate the ξ selected individuals of step S32e using the original fitness function (S32f),
- construct the neural network ensemble (S32g), and
- calculate the fitness of the remaining λ-ξ individuals using the neural network ensemble. (S32h)
- for the λ-ξ individuals, calculate the variance of output of the ensemble members. If the variance is larger than a given threshold, replace the fitness value with that of the individual closest to the cluster center

S33. Go to step S32 if the termination condition is not met

S34. Stop

Evolutionary optimization algorithm

Fig. 3

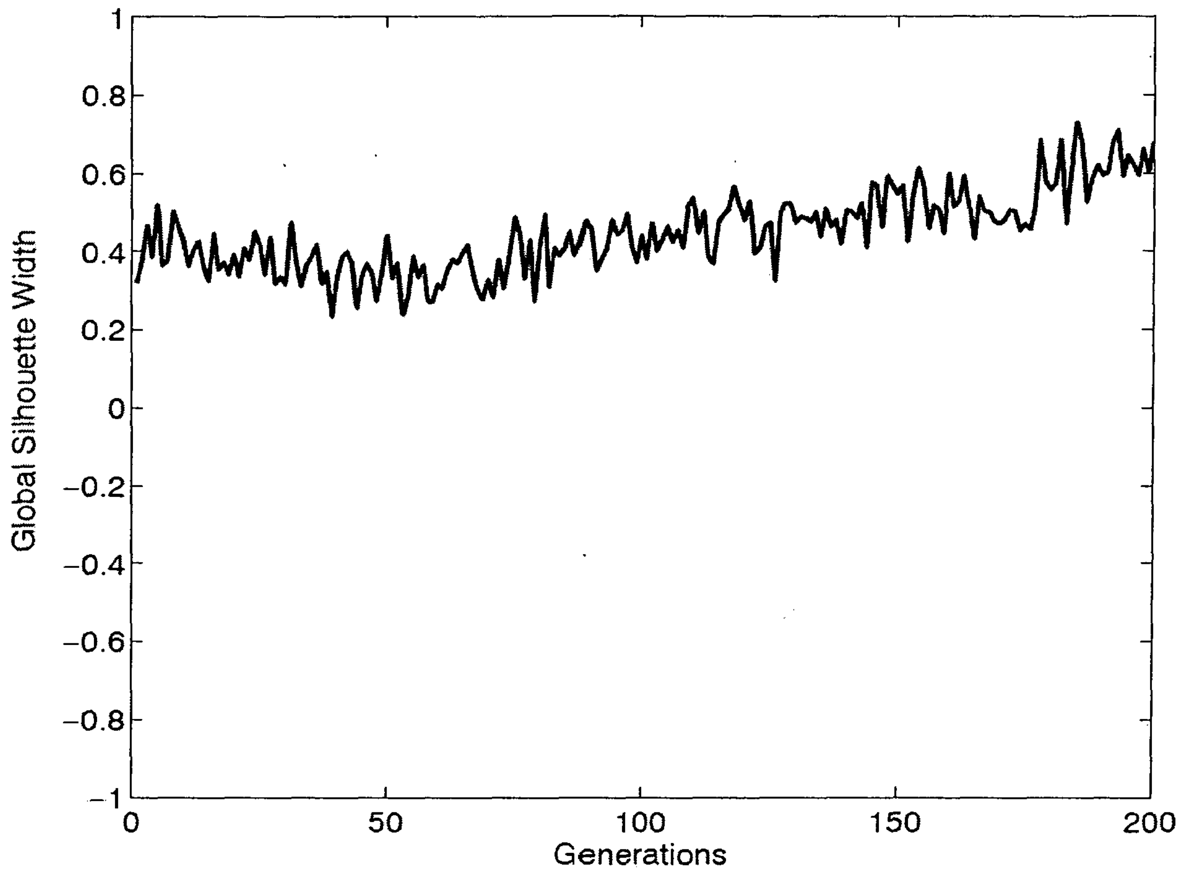
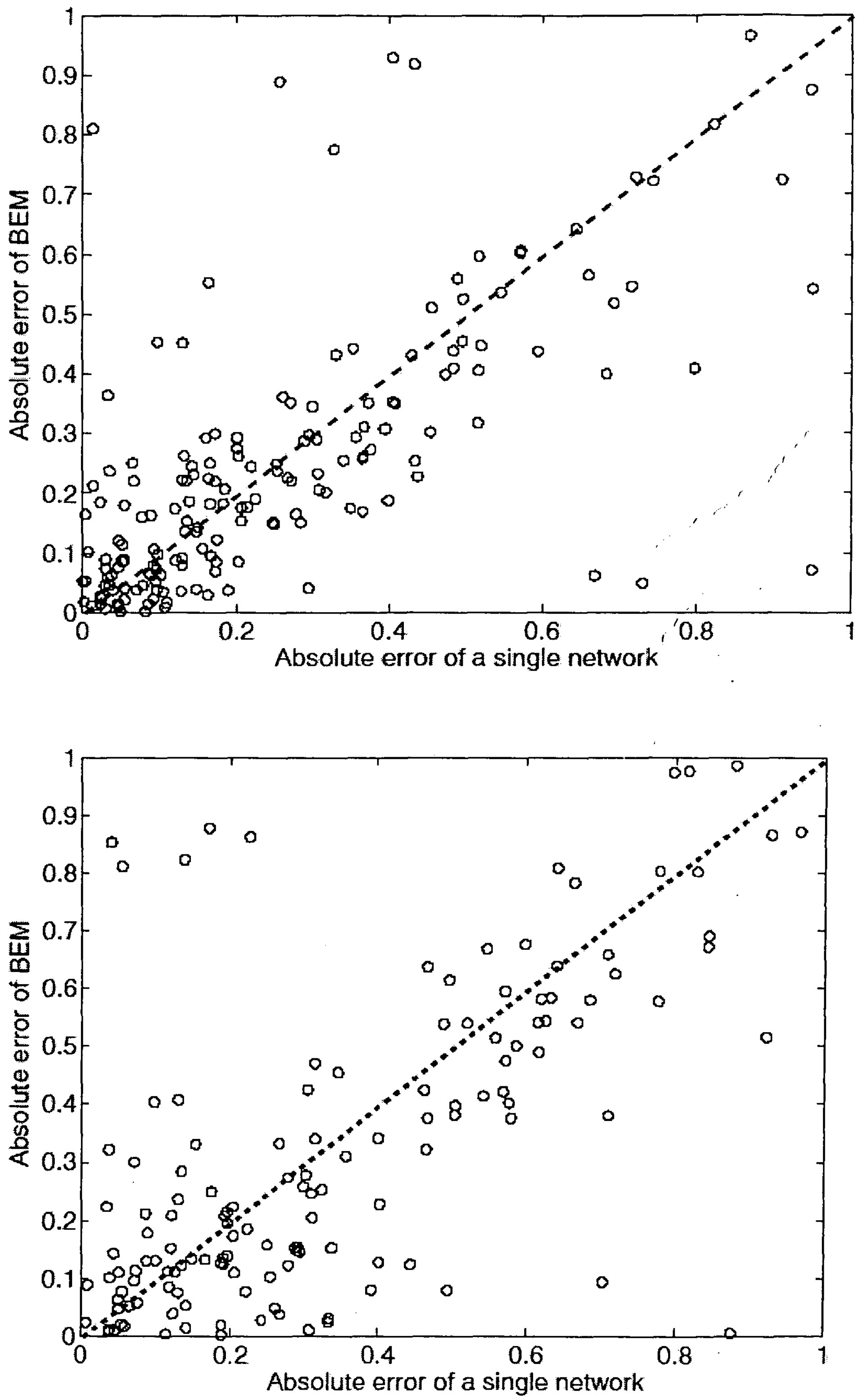

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
50
100
150
200
Generations
Global Silhouette Width

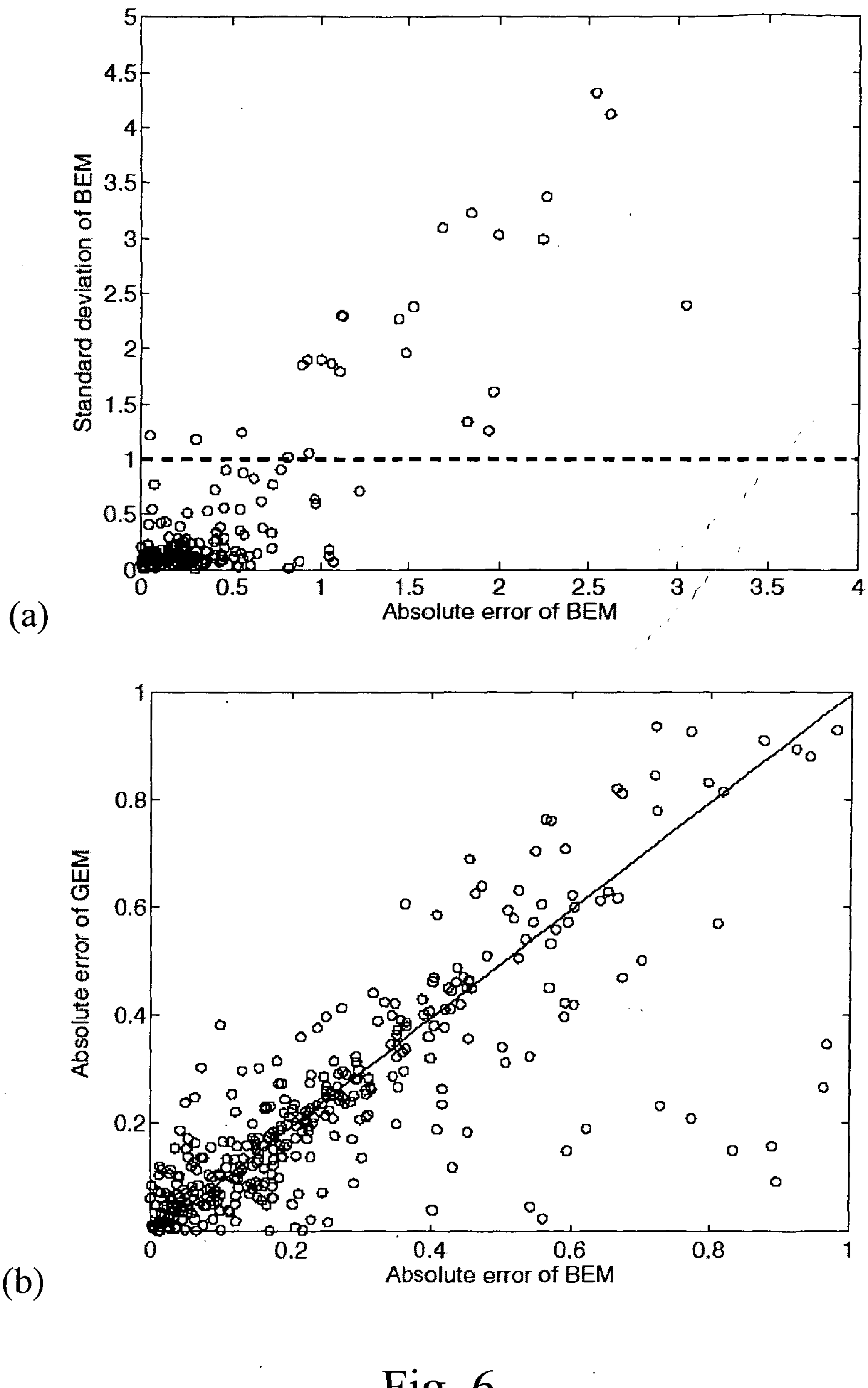
Fig. 4


Absolute error of BEM
Absolute error of a single network
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.4
0.6
0.8
1

(a)

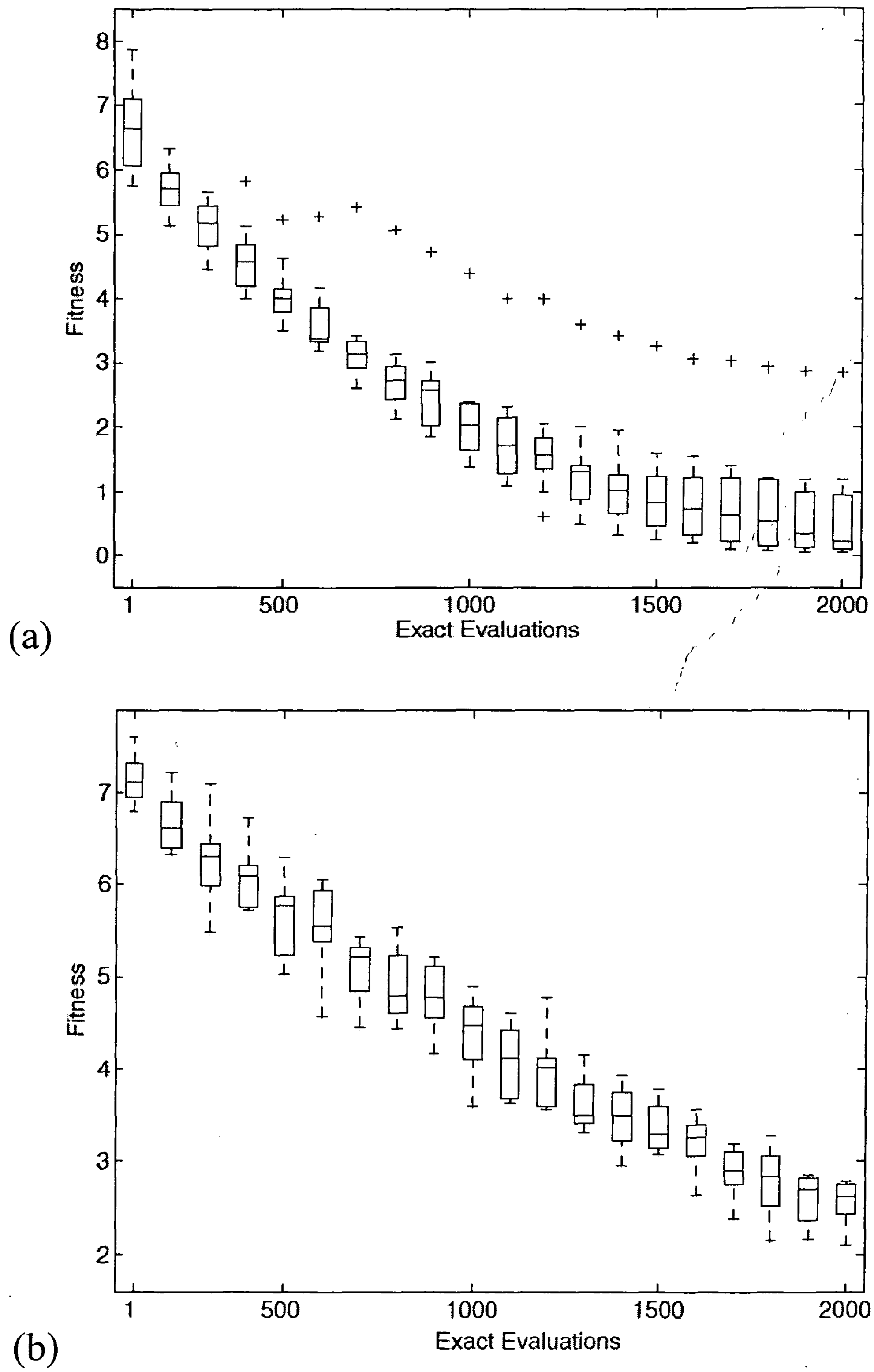
(b)

Fig. 5



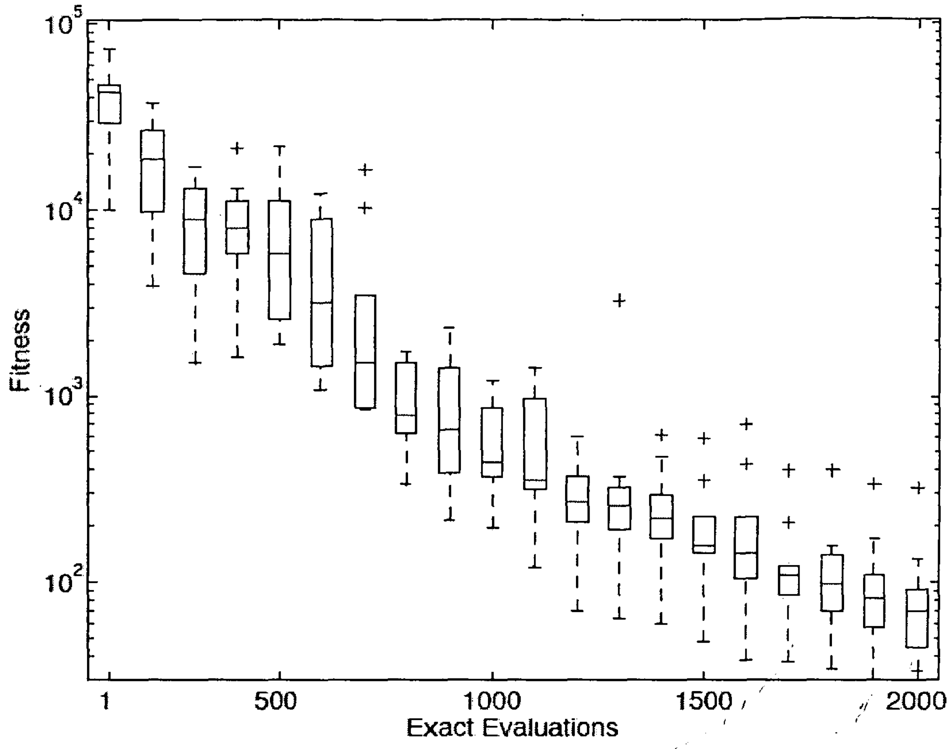
(a)
Standard deviation of BEM
Absolute error of BEM
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
(b)
Absolute error of GEM
Absolute error of BEM
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1

Fig. 6

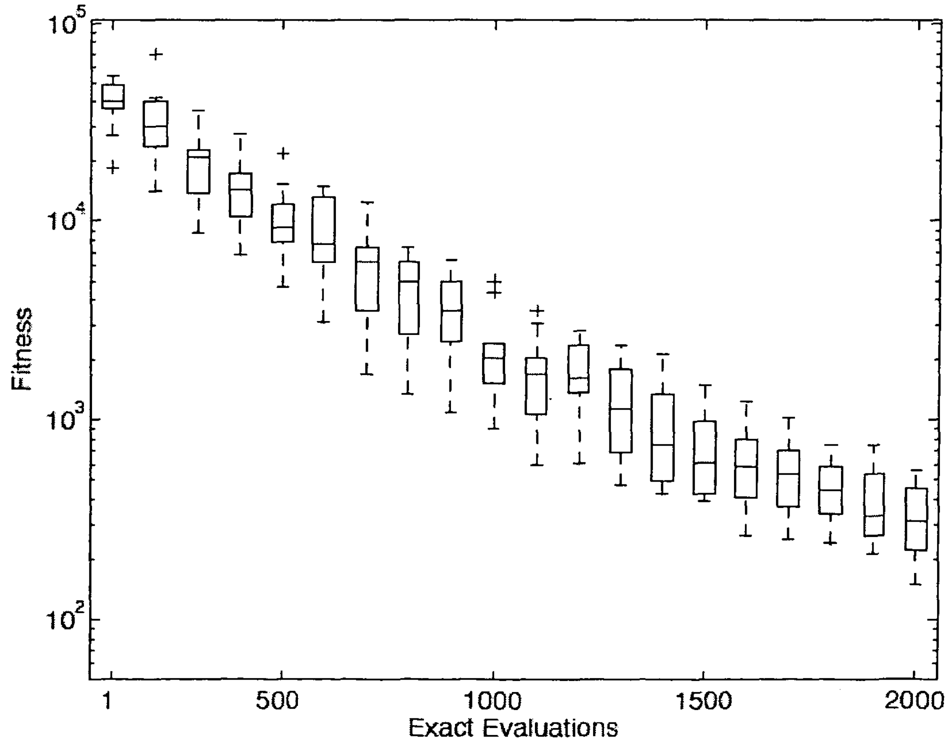


(a)
Fitness
8
7
6
5
4
3
2
1
0
1
500
1000
1500
2000
Exact Evaluations
(b)
Fitness
7
6
5
4
3
2
1
500
1000
1500
2000
Exact Evaluations

Fig. 7


Fitness
10^5
10^4
10^3
10^2
1
500
1000
1500
2000
Exact Evaluations

(a)


Fitness
10^5
10^4
10^3
10^2
1
500
1000
1500
2000
Exact Evaluations

(b)

Fig. 8

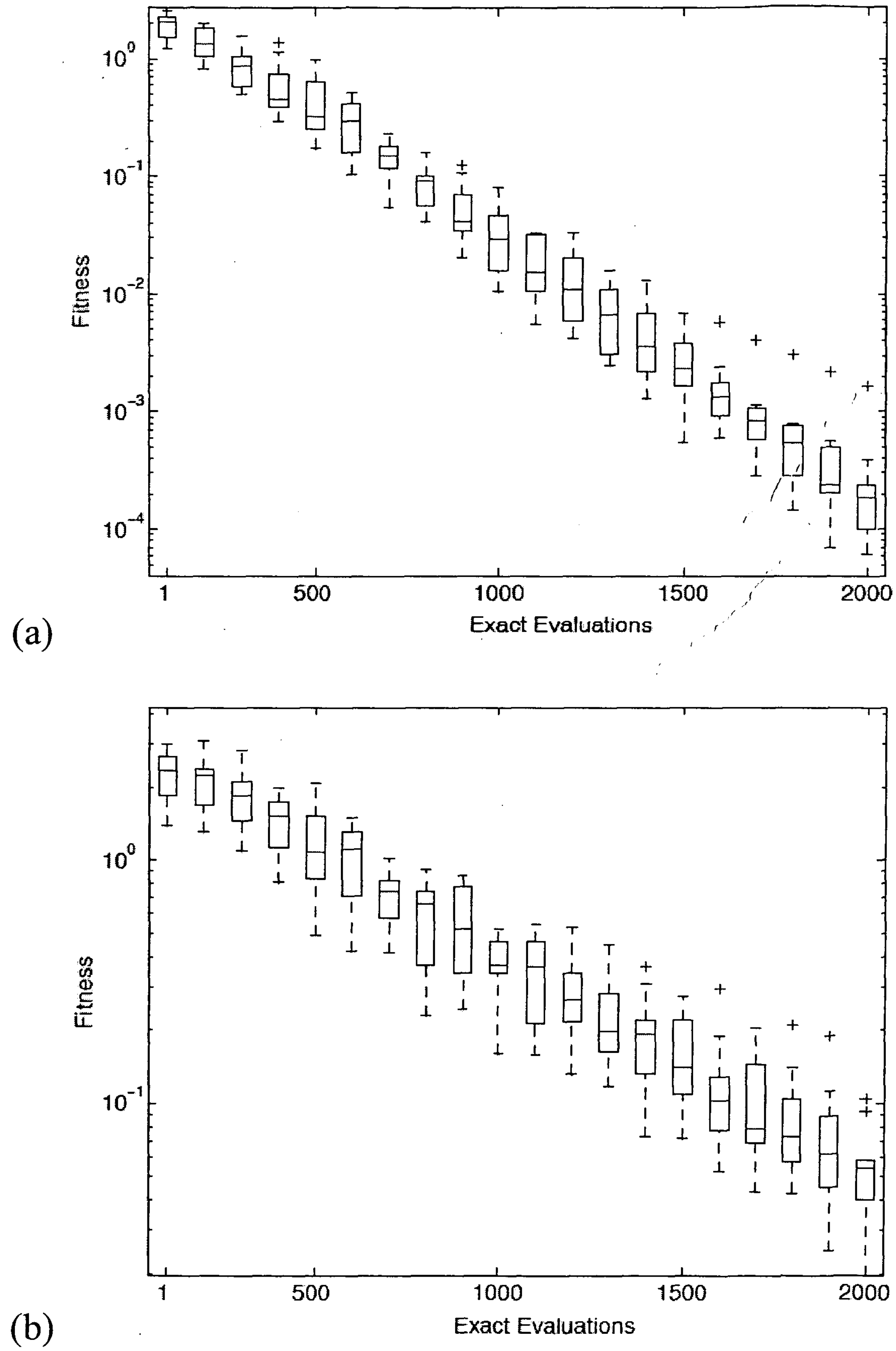

Fitness
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
1
500
1000
1500
2000
Exact Evaluations
(a)
Fitness
10^{0}
10^{-1}
1
500
1000
1500
2000
Exact Evaluations
(b)

Fig. 9

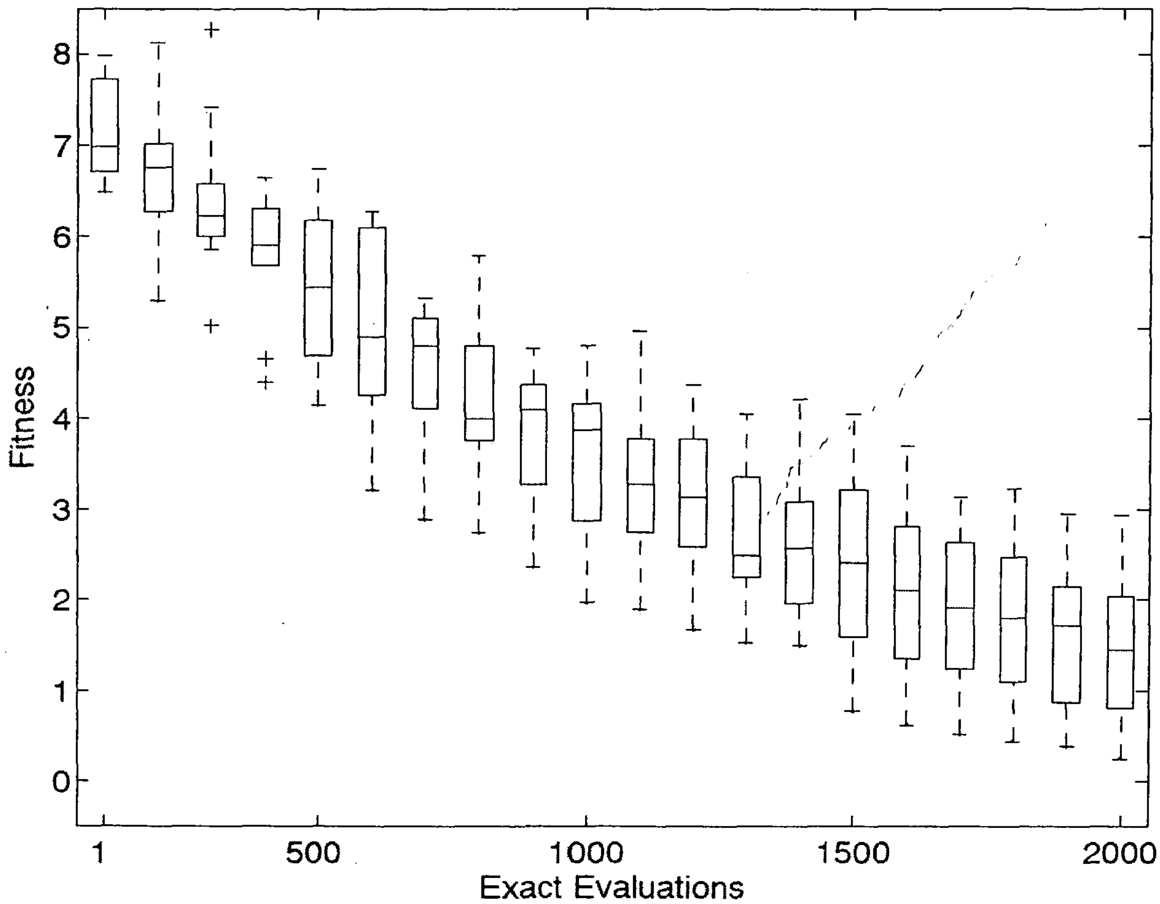


Fig. 10

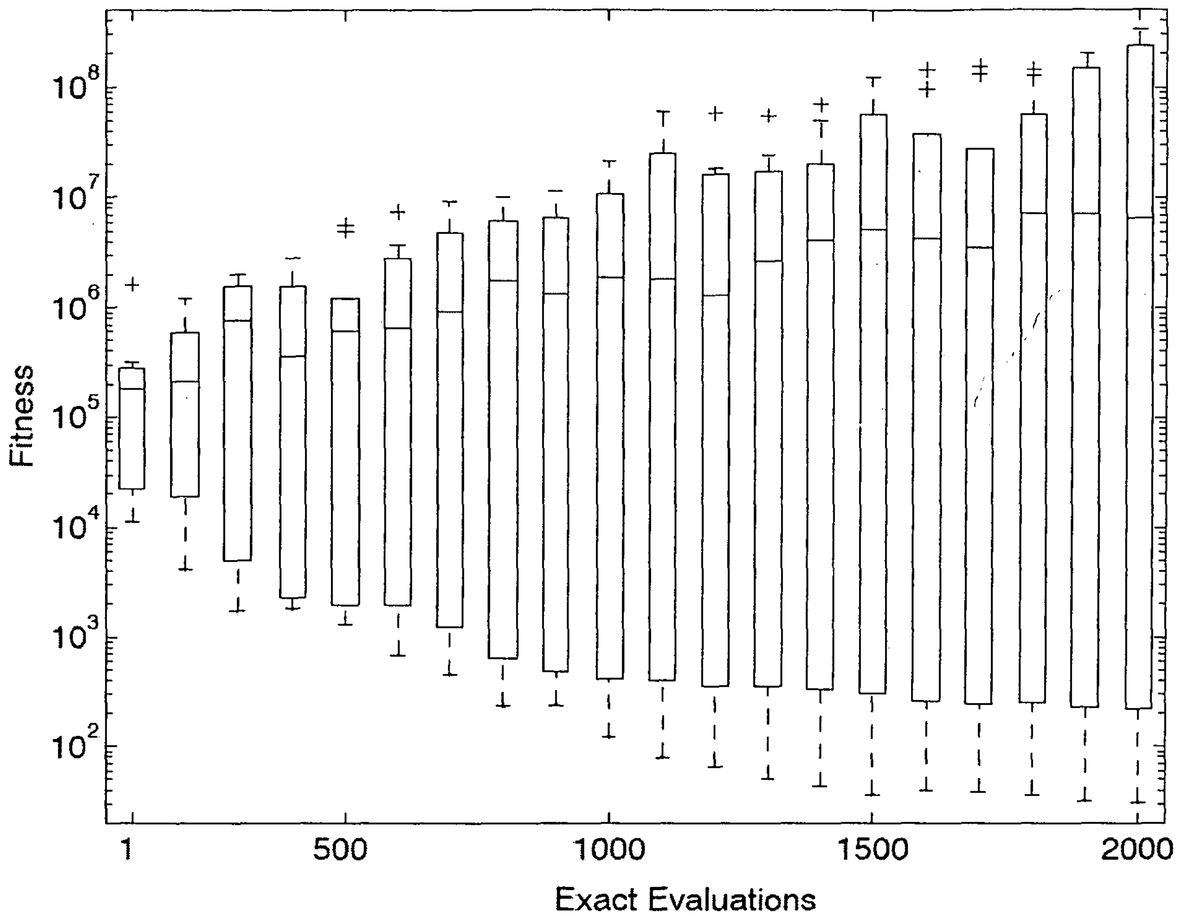
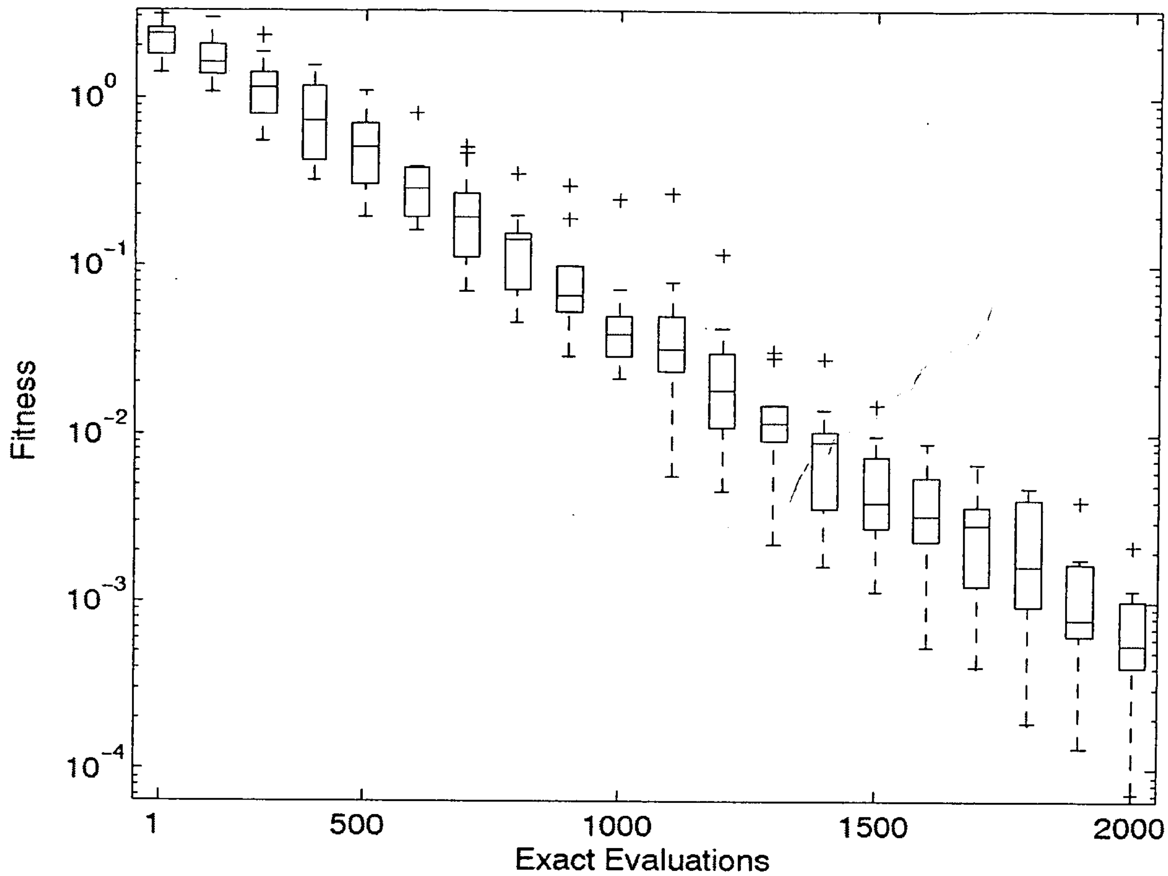

10^8
10^7
10^6
10^5
10^4
10^3
10^2
Fitness
1
500
1000
1500
2000
Exact Evaluations

Fig. 11



10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Fitness
1
500
1000
1500
2000
Exact Evaluations

Fig. 12